

# RFC 7440 : TFTP Window Size Option

Stéphane Bortzmeyer  
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 29 janvier 2015

Date de publication du RFC : Janvier 2015

<https://www.bortzmeyer.org/7440.html>

---

TFTP est un très vieux protocole de transfert de fichiers, surtout utilisé pour le démarrage de machines n'ayant pas les fichiers nécessaires en local, et devant les acquérir sur le réseau (par exemple via PXE). Reposant sur UDP et pas TCP comme tous les autres protocoles de transfert de fichiers, il peut être mis en œuvre dans un code de taille minimal, qui tient dans la mémoire morte. Par défaut, TFTP n'a pas de fenêtre d'envoi : chaque paquet doit faire l'objet d'un accusé de réception avant qu'on puisse envoyer le suivant. Ce RFC ajoute une option à TFTP qui permet d'avoir une fenêtre d'envoi, accélérant ainsi les transferts de fichier.

C'est le RFC 1350<sup>1</sup> qui est l'actuelle norme TFTP. Ce vieux RFC décrit le fonctionnement de base de TFTP en notant que « *each data packet must be acknowledged by an acknowledgment packet before the next packet can be sent* »<sup>1</sup>, ce qu'on nomme le "lock-step". Dit autrement, TFTP a une fenêtre de taille 1. Vu par le dissecteur de paquets de pcapr <<https://www.bortzmeyer.org/pcapr.html>>, cela donne une stricte alternance des données et des accusés de réception :

```
1. 192.168.0.1 » 192.168.0.13 tftp Write Request, File: rfc1350.txt\000, Transfer type: octet\000
2. 192.168.0.13 » 192.168.0.1 tftp Acknowledgement, Block: 0
3. 192.168.0.1 » 192.168.0.13 tftp Data Packet, Block: 1
4. 192.168.0.13 » 192.168.0.1 tftp Acknowledgement, Block: 1
5. 192.168.0.1 » 192.168.0.13 tftp Data Packet, Block: 2
6. 192.168.0.13 » 192.168.0.1 tftp Acknowledgement, Block: 2
7. 192.168.0.1 » 192.168.0.13 tftp Data Packet, Block: 3
8. 192.168.0.13 » 192.168.0.1 tftp Acknowledgement, Block: 3
```

---

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc1350.txt>

À l'époque du RFC 1350, il n'y avait aucun moyen de changer cela, TFTP n'ayant pas de mécanisme d'options. Mais le RFC 1782, remplacé depuis par le RFC 2347, lui en a donné un.

L'absence d'un mécanisme de fenêtre (ou, ce qui revient au même, le fait que la fenêtre soit de taille 1) serait dramatique pour des transferts sur l'Internet, où la latence <https://www.bortzmeyer.org/latence.html> est souvent élevée. Mais TFTP n'est quasiment utilisé que sur des LAN, où la latence est faible et les pertes de paquet rares. L'option `Blocksize` du RFC 2348, permettant des paquets plus grands que les 512 octets originaux, avait jusqu'à présent suffi à calmer les désirs de performance meilleure. Mais TFTP reste lent, et cette lenteur pose un problème lorsqu'on veut transférer de gros fichiers, comme une image Linux ou comme les PE de Microsoft utilisés par WDS/MDT/SCCM lorsqu'on démarre en PXE. Contrairement à ce que croient beaucoup de gens, TFTP est très loin d'être en voie d'extinction et il est donc justifié de chercher à l'améliorer. Le RFC se vante qu'avec cette option, TFTP peut aller aussi vite que les autres protocoles de transfert de fichier.

L'option elle-même est décrite dans la section 3. Le mécanisme d'option du RFC 2347 consiste en l'ajout de l'option à la fin du paquet `Read Request` ou `Write Request`. L'option comporte un nom (`window size` dans notre cas), et une valeur, qui est le nombre de blocs dans une fenêtre (rappelez-vous que, grâce à une autre option, celle du RFC 2348, un bloc ne fait pas forcément 512 octets). On peut envoyer autant de blocs qu'indiqué dans cette option, avant d'exiger un accusé de réception (la section 4 montre un schéma détaillé d'un transfert de données, avec une fenêtre de taille 4). Le récepteur des données accuse réception du dernier bloc (cela accuse implicitement réception des blocs précédents). Les accusés de réception TFTP indiquent en effet le numéro de bloc reçu (RFC 1350, dessin 5.3). Si un bloc se perd, l'émetteur des données s'en rendra compte en ne recevant d'accusé de réception que pour le bloc précédant le perdu, il pourra alors réémettre.

L'option `window size` aura dû être préalablement acceptée par le pair, via un OACK ("*Option Acknowledgment*", RFC 2347).

Pour choisir des bonnes valeurs de fenêtre, une expérience a été faite (section 5), transférant un fichier de 180 Mo, avec des blocs de taille 1 456 sur un Ethernet gigabit, entre deux PC. Avec la fenêtre de 1 (la valeur obligatoire, avant notre RFC), le transfert prend 257 secondes. Il baisse ensuite lorsque la taille de la fenêtre augmente (76 secondes pour une fenêtre de 4, 42 secondes pour une fenêtre de 16) puis plafonne, aux alentours de 35 secondes, d'une fenêtre de taille 64 (pour les grandes fenêtres, le risque accru qu'un paquet dans la fenêtre soit perdu compense l'avantage qu'il y a à ne pas attendre les accusés de réception, il ne sert donc à rien d'augmenter indéfiniment la taille de la fenêtre). Un transfert avec un protocole classique (SMB/CIFS) prend 23 secondes. On voit donc que TFTP peut presque atteindre la même valeur.

Cela ne veut pas dire qu'il faut forcément choisir aveuglément une fenêtre de 64 blocs dans tous les cas : la valeur optimale dépend du réseau entre les deux machines. Le RFC recommande donc de tester avant de choisir.

TFTP utilise UDP, qui n'a pas de mécanisme de contrôle de la congestion. Un émetteur de données TFTP doit donc suivre les règles de prudence du RFC 8085 (notamment sa section 3.1) pour ne pas surcharger le réseau (section 6 de notre RFC). Contrairement à TCP, TFTP n'offre pas de mécanisme permettant de réduire la taille de la fenêtre en cours de route. En cas de gros problème de perte de paquets, la seule solution est d'avorter le transfert (ce qu'on nomme un "*circuit breaker*") et de réessayer avec d'autres paramètres.

On trouve plein de traces TFTP sur `pcapr` <https://www.bortzmeyer.org/pcapr.html> mais aucune avec cette option. Parmi les mises en œuvres de TFTP, si le mécanisme d'options du RFC 2347

est souvent présent, ainsi que l'option de taille de bloc du RFC 2348, en revanche notre nouvelle option de taille des fenêtres ne semble pas encore souvent là. Parmi les programmes qui gèrent cette option : Serva (cité au paragraphe suivant), Node-tftp <<https://github.com/gagle/node-tftp>>...

Un bon article de l'auteur du RFC explique cette option, les options précédentes, et leur implémentation dans Serva <<http://www.vercot.com/~serva/>> : « *Advanced Topics on TFTP* » <<http://www.vercot.com/~serva/advanced/TFTP.html>> ».